Learning Latent Representations for Speech Generation and Transformation

Wei-Ning Hsu, Yu Zhang, James Glass

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{wnhsu,yzhang87,jrg}@csail.mit.edu

Abstract

An ability to model a generative process and learn a latent representation for speech in an unsupervised fashion will be crucial to process vast quantities of unlabelled speech data. Recently, deep probabilistic generative models such as Variational Autoencoders (VAEs) have achieved tremendous success in modeling natural images. In this paper, we apply a convolutional VAE to model the generative process of natural speech. We derive latent space arithmetic operations to disentangle learned latent representations. We demonstrate the capability of our model to modify the phonetic content or the speaker identity for speech segments using the derived operations, without the need for parallel supervisory data.

Index Terms: unsupervised learning, variational autoencoder, speech generation, speech transformation, voice conversion

1. Introduction

Speech waveforms have complex distributions that exhibit high variance due to factors that include linguistic content, speaking style, dialect, speaker identity, emotional state, environment, channel effects, etc. Understanding the influence of these factors on the speech signal is an important problem, which can be used for a wide variety of applications, including, but not limited to adaptation and data augmentation for speech recognition [1, 2], voice conversion [3, 4, 5], and speech compression [6]. However, most previous research has focused on handcrafting features to capture these factors, rather than learning these factors automatically through a probabilistic generative process.

Recently, there has been significant interest in deep probabilistic generative models, such as Variational Autoencoders (VAEs) [7] and Generative Adversarial Nets (GANs) [8]. Particularly, VAE addresses the intractability issue that occurs in even moderately complicated models such as Restricted Boltzmann machines (RBMs), which have been applied for voice conversion [9, 10, 11], and provides efficient approximated posterior inference of the latent factors. While there are many works investigating generative models for natural images [12, 13], little work has been done on learning speech generation with deep probabilistic generative models [14, 15].

In this paper, we adopt the VAE framework and propose a convolutional architecture to model the probabilistic generative process of speech to learn a latent representation. We present simple arithmetic operations in the latent space to demonstrate that such operations can decompose the latent representation into different attributes, such as speaker identity and linguistic content. By manipulating the latent representation, we also demonstrate an ability to perturb some aspect of the surface speech segment, for example the speaker identity, while keeping the remaining attributes fixed (e.g., linguistic content). To quantify the behavior of the latent representation modifications, an experiment is conducted to measure our ability to modify speaker characteristics without changing linguistic content, and vice versa. In addition, we perform an analysis to evaluate the model's ability to generate speech segments of different durations.

The rest of the paper is organized as follows. In Section 2, we briefly discuss related work. Our models and an analysis of latent representations are detailed in Section 3 and 4. Data preparation is explained in Section 5. In Section 6, we show the experimental results. Finally, we conclude our work and discuss our future research plans on this topic in Section 7.

2. Related Work

Recent research on speech and audio generation has made remarkable progress on directly utilizing time-domain speech signals. WaveNet [16] introduces the one-dimensional dilated causal convolutional model, where the effective receptive field grows exponentially wide with the depth by using exponentially growing dilation factors with the depth. A different model called SampleRNN [17] presented a multi-scale recurrent neural network, where each layer is operated at different clock rates and each sample is generated conditioned on all the previous samples. Both models focused on generating high quality audio segments by predicting the next sample given the preceding samples, instead of learning latent representations for the entire audio segments using probabilistic generative models.

While VAEs have been widely applied for image generation, there has been less speech research on this topic. A VAE-based framework was used in [18] to extract both frame-level and utterance-level features that were used in combination with other features for robust speech recognition. A fully-connected VAE was used in [14] to learn a frame-level latent representation, and evaluated using a Gaussian diffusion process to generate and concatenate multiple samples that varied smoothly in time.

3. Model

3.1. Variational Autoencoder

Variational autoencoders [7] define a probabilistic generative process between observation $\boldsymbol{x}$ and latent variable $\boldsymbol{z}$ as follows: $\boldsymbol{z} \sim p_{\boldsymbol{\theta}^*}(\boldsymbol{z})$ and $\boldsymbol{x} \sim p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{z})$, where the prior $p_{\boldsymbol{\theta}^*}(\boldsymbol{z})$ and the conditional likelihood $p_{\boldsymbol{\theta}^*}(\boldsymbol{x}|\boldsymbol{z})$ are from a probability distribution family parameterized by $\boldsymbol{\theta}$. In an unsupervised setting, we are only given a dataset $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$, so the true value of $\boldsymbol{\theta}^*$, as well as the latent variable $\boldsymbol{z}$ for each observation $\boldsymbol{x}$ in this process are unknown.

We are often interested in knowing the marginal likelihood of the data $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, or the posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$; however, both require computing the intractable integral $\int p_{\boldsymbol{\theta}}(\boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})d\boldsymbol{z}$.

To solve this problem, VAEs introduce a recognition model $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, which approximates the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$. We can therefore rewrite the marginal likelihood as:

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= D_{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) \\ &\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}) \\ &= -D_{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})], \quad (1)\end{aligned}$$

where $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x})$ is the variational lower bound we want to optimize with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$.

In the VAE framework we consider here, both the recognition model $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and the generative model $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ are parameterized using diagonal Gaussian distributions, of which the mean and the covariance are computed with a neural network. The prior is assumed to be a centered isotropic multivariate Gaussian $p_{\boldsymbol{\theta}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$, that has no free parameters.

In practice, the expectation in (1) is approximated by first drawing L samples from $\boldsymbol{z}^l \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, and then computing $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] \simeq \frac{1}{L}\sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}^l)$. To yield a differentiable network after sampling, the reparameterization trick [7] is used. Suppose $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\sigma}_{\boldsymbol{z}}^2\boldsymbol{I})$, after reparameterizing we have $\boldsymbol{z} = \boldsymbol{\mu}_{\boldsymbol{z}} + \boldsymbol{\sigma}_{\boldsymbol{z}} \odot \boldsymbol{\epsilon}$, where $\odot$ denotes an element-wise product, and vector $\boldsymbol{\epsilon}$ is sampled from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and treated as an additional input.

3.2. Proposed Model Architecture

In this work, our goal is to learn latent representations of speech segments to model the generation process. We let the observed data $\boldsymbol{x}$ be a sequence of frames of fixed length. The learned latent variable $\boldsymbol{z}$ is therefore supposed to encode the factors that result in the variability of speech segments, such as the content being spoken, speaker identity, and channel effect.

As mentioned earlier, a VAE is composed of two networks: a recognition network, and a generative network. The recognition network takes a speech segment as input and predicts the mean $\boldsymbol{\mu}_{\boldsymbol{z}}$ and the log-variance $\log \boldsymbol{\sigma}_{\boldsymbol{z}}^2$ that parameterize the posterior distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. A speech segment is treated as a two dimensional image of width T and height F; however, unlike natural images, speech segments are only translational invariant to the time axis. Therefore, similar to [19], 1-by-F filters are applied at the first convolutional layer, and w-by-1 filters at following layers. As suggested in [12], instead of pooling, we use stride size > 1 for down-sampling along the time axis. The output from the last convolutional layer is flattened and fed into fully connected layers before going to the Gaussian parameter layer modeling the latent variable $\boldsymbol{z}$. See Table 1 for a summary.

The generative network takes sampled $\boldsymbol{z}$ as input, and predicts the mean $\boldsymbol{\mu}_{\boldsymbol{x}}$ as well as the log-variance $\log \boldsymbol{\sigma}_{\boldsymbol{x}}^2$ of the observed data. Here we use symmetric architectures to the corresponding recognition network.

	Conv1	Conv2	Conv3	Fc1	Gauss
#filters/units	64	128	256	512	128
filter size	1xF	3x1	3x1	-	-
stride	(1,1)	(2,1)	(2,1)	-	-

Table 1: *Recognition network architecture. Conv refers to convolutional layers, Fc refers to fully connected layers, and Gauss refers to the Gaussian parametric layer modeling $\boldsymbol{z}$*

Different choices for the activation function were investigated. No activation is applied to Gaussian parameter layers, since the mean and the log-variance are unbounded for both $\boldsymbol{x}$ and $\boldsymbol{z}$. For other layers, we use tanh because the unbounded rectifier linear units led to overflow of the KL-divergence and conditional likelihood. Batch normalization is applied to every layer except for the Gaussian parameter layer.

4. Latent Representation Analysis

In this section, we examine how to decompose speech attributes from the learned latent representations. Here, we use a to denote the attribute and r to denote the value of the attribute.

4.1. Deriving Latent Attribute Representations

The first assumption we make is that conditioning on some attribute a being r, such as the *phone* being */ae/*, the prior distribution of $\boldsymbol{z}$ is also a Gaussian; in other words, $p(\boldsymbol{z}; a = r) = N(\boldsymbol{z}; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$. We therefore define $\boldsymbol{\mu}_r$ as the *latent attribute representation* for r. Let $\boldsymbol{X}_r = \{\boldsymbol{x}_r^{(i)}\}_{i=1}^{N_r}$ be a subset of $\boldsymbol{X}$ where the attribute a of each instance is r. We can then estimate $\boldsymbol{\mu}_r$ as follows:

$$\begin{aligned}\boldsymbol{\mu}_r &= \mathbb{E}_{p_{\theta}(\boldsymbol{z};r)}[\boldsymbol{z}] = \int_{\boldsymbol{z}}\int_{\boldsymbol{x}} \boldsymbol{z} p_{\theta}(\boldsymbol{z}|\boldsymbol{x}; r) p_{\theta}(\boldsymbol{x}; r) \\ &\approx \int_{\boldsymbol{z}}\int_{\boldsymbol{x}} \boldsymbol{z} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}; r) p_{\boldsymbol{\theta}}(\boldsymbol{x}; r) \approx \frac{1}{N_r}\sum_{i=1}^{N_r}\int_{z} \boldsymbol{z} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_r^{(i)}) \\ &\approx \frac{1}{N_r}\sum_{i=1}^{N_r}(\frac{1}{J}\sum_{j=1}^{J}\boldsymbol{z}^{(i,j)}), \quad (2)\end{aligned}$$

where $\boldsymbol{z}^{(i,j)} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_r^{(i)})$. This results in averaging the J sampled latent representations of each instance in $\boldsymbol{X}_r$. Furthermore, let $\bar{\boldsymbol{z}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{z}_i$, since $p(\boldsymbol{z})$ is a log-concave function, it is guaranteed that $p(\bar{\boldsymbol{z}}) > \min_{\boldsymbol{z}_i} p(\boldsymbol{z}_i)$. Therefore VAE should be able to generate reasonable speech-like segment from $\bar{\boldsymbol{z}}$ if all the $p(\boldsymbol{z}_i)$ have high values.

4.2. Arithmetic Operations to Modify Speech Attributes

Here we make the second assumption: let there be K independent attributes that affect the realization of speech, each attribute a_k is then modeled using a subspace Z_{a_k}, where $Z = \cup_{k=1}^{K} Z_{a_k}$ and $Z_{a_k} \perp Z_{a_{k'}}$ if $k \neq k'$. Hence, the latent representation can be decomposed into K orthogonal latent attribute representations $\boldsymbol{z}_{a_1}, \boldsymbol{z}_{a_2}, \cdots, \boldsymbol{z}_{a_k}$, where $\boldsymbol{z}_{a_k} \in Z_{a_k}$ and $\boldsymbol{z} = \sum_{k=1}^{K}\boldsymbol{z}_{a_k}$. Combining the aforementioned assumption of the conditioned prior of $\boldsymbol{z}$, we can next derive the latent space arithmetic operations to modify the speech attributes.

Suppose we want to modify the attribute a_k, for example the speaker identity, of a speech segment $\boldsymbol{x}^{(i)}$, from being speaker r_s to being speaker r_t. Given the latent attribute representations $\boldsymbol{\mu}_{r_s}$ and $\boldsymbol{\mu}_{r_t}$ for speaker r_s and r_t respectively, the *latent attribute shift* $\boldsymbol{v}_{r_s \to r_t}$ is computed as: $\boldsymbol{v}_{r_s \to r_t} = \boldsymbol{\mu}_{r_t} - \boldsymbol{\mu}_{r_s}$. We can then modify the speech $\boldsymbol{x}^{(i)}$ as follows:

$$\boldsymbol{z}^{(i)} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \quad (3)$$

$$\boldsymbol{z}_{mod}^{(i)} = \boldsymbol{z}^{(i)} + \boldsymbol{v}_{r_s \to r_t} \quad (4)$$

$$\boldsymbol{x}_{mod}^{(i)} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_{mod}^{(i)}), \quad (5)$$

which does not modify latent attribute representations other than $\boldsymbol{z}_{a_k}^{(i)}$, because $\boldsymbol{v}_{r_s \to r_t} \perp \boldsymbol{z}_{a_{k'}}^{(i)}$ for $k' \neq k$.

5. Data

5.1. TIMIT

The TIMIT acoustic-phonetic corpus [20, 21] contains broadband recordings of phonetically-balanced read speech. A total of 6300 utterances (5.4 hours) are presented with 10 sentences from each of 630 speakers, of which approximately 70% are male and 30% are female. Each utterance comes with manually time-aligned phonetic and word transcriptions, as well as a 16-bit, 16kHz speech waveform file. We follow Kaldi's TIMIT recipe to split train/dev/test sets and exclude dialect sentences (SA), with 462/50/24 non-overlapping speakers in each set respectively. Phonetic transcriptions are based on 58 phones, excluding silence phones.

5.2. Data Preprocessing

We consider two types of frame representations: magnitude spectrum in dB (Spec) and filter banks (FBank). For both features, we first apply a 25ms Hanning window with 10ms shift, and then compute the short time Fourier transform coefficients with flooring at -20dB. For FBank features, 80 Mel-scale filter banks that match human perceptual sensitivity are applied, which preserves more detail at lower frequency regions.

We investigate two different segment lengths: 200ms and 1s, which correspond to 20 frames and 100 frames, and are referred to as *syllable-level* and *word-level* datasets, respectively.

6. Experimental Results

6.1. Experiment Setups

All models were trained with stochastic gradient descent using a mini-batch size of 128 without clipping to minimize the negative variational lower bound plus an $L2$-regularization with weight 10^{-4}. The Adam [22] optimizer is used with $\beta_1 = 0.95$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and initial learning rate of 10^{-3}. Training is terminated if the lower bound on the development set does not improve for 10 epochs. To compare with VAE, we also train an autoencoder (AE) with the same proposed model architecture except for the Gaussian latent variable layer, which is replaced with a fully-connected layer of 128 hidden units[1].

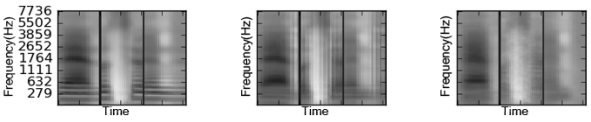
6.2. Latent Attribute Representation

In Figure 1, we show the results of reconstructing from latent attribute representations of three phones, /ae/, /th/, and /n/, using VAE and AE respectively, based on the derivation in Section 4.1. As a baseline, we also show the results of averaging filter bank features. The VAE preserves more harmonic structure and clearer spectral envelope, while the AE and the Fbank are more blurred. It is worth noting that AE also shows unnatural frequent vertical stripe artifacts.

6.3. Modifying Attributes of Speech

To assess the orthogonality-between-attributes assumption, we sampled six speakers, three males and three females, denoted by $\{m,f\}\_spk[i]$, and ten phones, including vowels, stops, fricatives, and nasals, to compute three latent speaker representations and ten latent phone representations. Figure 2 plots the cosine similarities between these representations. From the figure, we can observe that off-diagonal blocks have low cosine similarities, which indicates that latent speaker representations and latent phone representations reside in orthogonal latent subspaces. Second, different latent phone representations also cluster according to the phonetic characteristics, which suggests the latent phone subspace may be further divided.

[1]Both VAE and AE models show reasonable reconstruction performance on both Fbank and Spec. We do not show the reconstructed features in this section due to space limitations.



(a) VAE (b) AE (c) Fbank

Figure 1: *Comparison between VAE, AE and Fbank on averaging representations of /ae/, /th/, and /n/ from left to right. Each segment is 200ms long.*

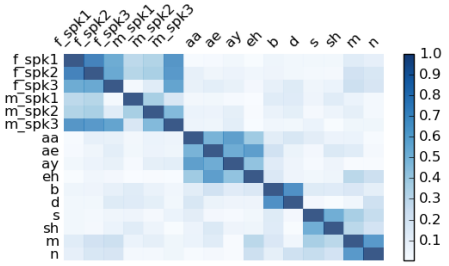


Figure 2: *Cosine similarities of latent attribute representations.*

We next explored modifying the phone and speaker attributes using the derived operations in Section 4.2.[2] Figure 3 shows an example of drawing 10 instances of the phone /aa/ and transforming them to /ae/ using the latent attribute shift $\boldsymbol{v}_{aa \to ae}$. We can clearly observe that the second formant F_2, marked with red boxes,[3] of each instance goes up after modification, because it is being changed from a back vowel to a front vowel. On the other hand, the harmonics of each instance, which are closely related to the speaker identity, maintain roughly the same.

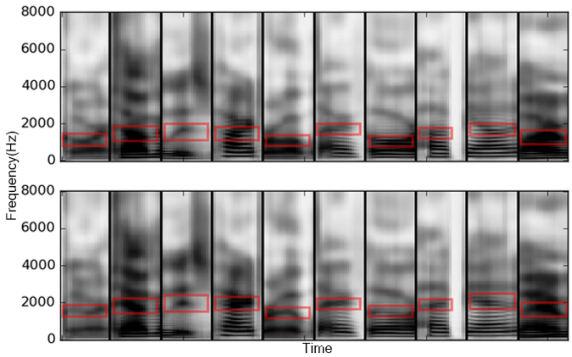


Figure 3: *Modify the phone from /aa/ (top) to /ae/ (bottom). Each segment is 200ms long.*

Figure 4 illustrates modifying 10 instances from a female speaker *falk0* to a male speaker *madc0* with the latent attribute

[2]More sound examples can be found at: `http://people.csail.mit.edu/wnhsu/vae_speech`

[3]Best viewed in color

shift $\boldsymbol{v}_{falk0 \to madc0}$. The harmonics (horizontal stripes) decrease after modification, while the spectrum envelope remains the same, indicating that the phonetic content is not changed.

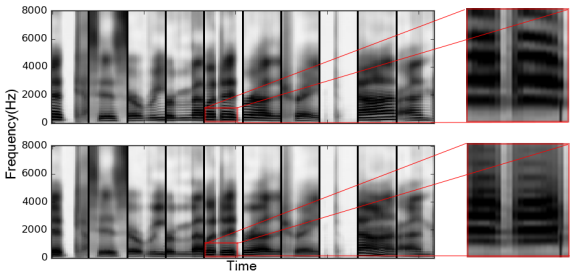


Figure 4: *Modify from a female (top) to a male (bottom). Each segment is 200ms long.*

In an attempt to quantify our latent attribute perturbation, we trained convolutional phonetic and speaker classifiers so that we could measure the difference of the posterior of each attribute before and after modification. The 58-class phone classifier achieves a test accuracy of 72.2%, while the 462-class speaker classifier achieves a test accuracy of 44.2%.

The shifts in posterior distributions of the phone and speaker classifications on the modified data are shown in Table 2. The upper half of the table contains results for speech segments that were transformed from /aa/ to /ae/. The first row shows that the average /aa/ posterior was 34% while the average correct speaker posterior was 51%. The second row shows that after modification to an /ae/, the average phone posteriors shift dramatically to be 30% /ae/, while slightly degrading the average correct speaker posterior.

The lower part of the table shows the results of speech segments that had speaker identity modified from speaker 'falk0' to 'madc0'. The third row shows an average speaker posterior of 44% for 'falk0' in the unmodified samples, while the average correct phone posterior was 55%. After modification we see that the average speaker posterior has shifted to be 29% 'madc0' while slightly degrading the average correct phone posterior.

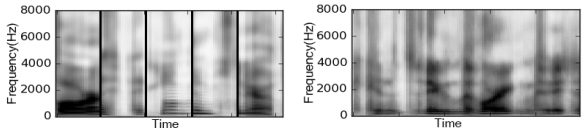
		/aa/	/ae/	ori. spk.
Modify Phone	before	34.06%	0.45%	50.78%
	after	0.24%	29.73%	41.66%
		falk0	madc0	ori. phone
Modify Speaker	before	44.48%	0.02%	54.61%
	after	3.11%	28.71%	48.71%

Table 2: *Average posteriors over 10 instances of source, target, and fixed attributes before and after modification.*

6.4. Random Sampling from the Latent Space

One of the advantages of VAEs is that the prior $p_\theta(z)$ is assumed to be a centered isotropic Gaussian, which enables us to sample latent vectors and reconstruct speech-like segments. Here, we investigate the syllable-level and word-level datasets.

Figure 5 (a) shows five random samples from the syllable-level model, which look and sound reasonable; however, we observe that random samples drawn from the word-level model are less natural because of excessive closures (vertical stripes), as shown in Figure 5 (b). The failure from drawing random samples implies that there is discrepancy between the assumed prior and the true prior. We hypothesize that because per-dimension KL-divergence values are computed, and correlations among dimensions are not penalized, the covariance matrix of the true prior may not be diagonal. We estimate the covariance matrix of the true prior by sampling the latent representations of the entire test set and compute the full covariance matrix. Figure 6 compares the syllable model and the word model on the sum of off-diagonal covariance scale for each dimension. We can observe that the word-level model has higher correlations between different dimensions than the syllable-level model.



(a) Syllable-level (b) Word-level

Figure 5: *Random samples drawn from models trained with syllable-level and word-level dataset. The segments in (a) are 200ms, and the segment in (b) is 1s.*

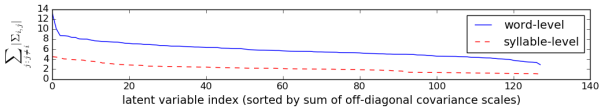
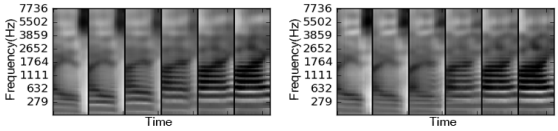


Figure 6: *Comparison of sum of off-diagonal covariance scales for each dimension for the syllable and word-level dataset.*

6.5. Walking in the Latent Space

Finally, we explore the operation of interpolation in the latent space between speech segments. Since $p(\boldsymbol{z})$ is log-concave, the interpolated $\boldsymbol{z}_{int} = \alpha \boldsymbol{z}_a + (1-\alpha)\boldsymbol{z}_b$, where $\alpha \in [0, 1]$, would have $p(\boldsymbol{z}_{int}) \geq \min(p(\boldsymbol{z}_a), p(\boldsymbol{z}_b))$. Therefore it should also generate reasonable speech-like segments. Figure 7 shows the transition between a male /ey/ to a female /ay/ using VAE and AE respectively. For VAE, we can clearly observe the pitch shifting and the formant contour transforming; however for AE it is more akin to interpolation in the raw feature space, where the magnitude of one segment goes down as the other goes up.



(a) VAE (b) AE

Figure 7: *Interpolation in the latent space using VAE and AE. Each segment is 200ms long.*

7. Conclusions and Future Work

In this paper, we present a convolutional VAE to model the speech generation process, and learn latent representations for speech in an unsupervised framework. The abilities to decompose the learned latent representations and modify attributes of speech segments are demonstrated qualitatively and quantitatively. For future work, we plan to extend to hierarchical recurrent models in order to capture information at different time scales, and generate speech of variable lengths.

8. References

[1] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *ICML workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.

[2] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 9, pp. 1469–1477, 2015.

[3] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *ICASSP*, 1998.

[4] Y. Stylianou, "Voice transformation: A survey." in *ICASSP*. IEEE, 2009, pp. 3585–3588.

[5] T. Toda, Y. Ohtani, and K. Shikan, "Eigenvoice conversion based on gaussian mixture model," in *Interspeech*, 2006, pp. 2446–2449.

[6] D. Wong, B. Juang, and D. Cheng, "Very low data rate speech compression with LPC vector and matrix quantization," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'83.*, vol. 8. IEEE, 1983, pp. 65–68.

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[9] Z. Wu, E. S. Chng, and H. Li, "Conditional restricted boltzmann machine for voice conversion," in *ChinaSIP*, 2013.

[10] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using speaker-dependent conditional restricted boltzmann machine," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, p. 8, 2015.

[11] T. Nakashika, T. Takiguchi, Y. Minami, T. Nakashika, T. Takiguchi, and Y. Minami, "Non-parallel training in voice conversion using an adaptive restricted boltzmann machine," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 24, no. 11, pp. 2032–2045, Nov. 2016.

[12] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[13] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *arXiv preprint arXiv:1512.09300*, 2015.

[14] M. Blaauw and J. Bonada, "Modeling and transforming speech using variational autoencoders," *Interspeech 2016*, pp. 1770–1774, 2016.

[15] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–6.

[16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR abs/1609.03499*, 2016.

[17] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," *arXiv preprint arXiv:1612.07837*, 2016.

[18] S. Tan and K. C. Sim, "Learning utterance-level normalisation using variational autoencoders for robust automatic speech recognition," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 43–49.

[19] D. Harwath and J. R. Glass, "Learning word-like units from joint audio-visual analysis," *arXiv preprint arXiv:1701.07481*, 2017.

[20] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[21] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351–356, 1990.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.